<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Integer programming formulation for convoy movement problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Kumar, P. N. Ram.; Narendran, T. T.</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2008</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/6153">http://hdl.handle.net/10220/6153</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2008 Inderscience Publishers</td>
</tr>
</tbody>
</table>

This is the author created version of a work that has been peer reviewed and accepted for publication by International journal of intelligent defence support systems, Inderscience Publishers. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: [DOI: http://dx.doi.org/10.1504/IJIDSS.2008.023006].
Integer programming formulation for convoy movement problem

P.N. Ram Kumar* and T.T. Narendran

Department of Management Studies,
Indian Institute of Technology Madras,
Chennai – 600 036, India

E-mail: pnram_kumar@yahoo.com
*Corresponding author

E-mail: ttn@iitm.ac.in

Abstract: This paper reports a study of the strategic network problem of routing military convoys between specific origin and destination pairs. Known as the Convoy Movement Problem (CMP), this problem is formulated as an integer linear program. The proposed mathematical model is evaluated on the basis of average number of iterations and average CPU times. LP-based lower bounds and heuristic based upper bounds were generated and used for evaluating the proposed model, particularly for large problem instances for which optimal solutions could not be obtained.

Keywords: military convoys; routing; scheduling; integer program; heuristics.


Biographical notes: P.N. Ram Kumar is currently a doctoral student at the Department of Management Studies, Indian Institute of Technology Madras. He received his Bachelor’s Degree in Mechanical Engineering from J.N.T. University, Hyderabad and a Master’s Degree in Industrial Engineering from the PSG College of Technology, Coimbatore. His research interests include combinatorial optimisation problems, reliability engineering and logistics management.

T.T. Narendran is with the Faculty of Operations Management at the Indian Institute of Technology Madras. He has done extensive research in the area of Cellular Manufacturing Systems and Flexible Manufacturing Systems. His current research interests include vehicle routing, logistics management, rail traffic management and ergonomics. He has published extensively in leading research journals such as the International Journal of Production Research and the European Journal of Operational Research.
1 Introduction

The problem of routing and scheduling military convoys between specific origin and destination pairs is known as the Convoy Movement Problem (CMP). Problems analogous to CMP include routing of Automated Guided Vehicles (AGV) in FMS environment (Batta et al., 1993), movement of luggage from different flights along a common automated transportation system to various pickup points (Bovet et al., 1991), strategic level routing of hazardous materials (Iakovou et al., 1999) and scheduling of trains on a single track (Higgins et al., 1996).

A defence establishment would typically need to move large number of personnel and equipment from their home bases to regions of conflict, threat or crisis, as rapidly as possible. In the process of movement, each military unit must be deployed as a convoy and certain precedence relationships must be followed among different units. Armoured Fighting Vehicles (AFVs) such as tanks and armoured personnel carriers are transported using specially designed trucks called transporters. At the time of deployment, convoys may use multiple modes of transportation. Road, Rail, Waterways and Airways are possible modes of transport. Convoys continue to move till they reach their respective destinations without halting en-route. They are allowed to cross/pass each other at junctions. Crossing or passing on roads is called a conflict and is not permitted to occur. This is because the roads along which convoys travel may not have the load bearing ability and width to accommodate two convoys at a time. The same is equally applicable in rail transportation also, as most of the towns are connected by single line railway networks.

The CMP can be stated as follows: Given a set of military convoys, and the origin and destination pair associated with each convoy, determine the routes and schedules for all the convoys on a multi-modal transportation network so as to minimise the sum of arrival times of the convoys at their respective destinations such that no convoy stops en-route and no two convoys cross or overtake each other along the roads.

The remainder of the paper is organised as follows. Section 2 provides a brief literature review. Section 3 gives a detailed description of the proposed mathematical formulation followed by a description of generation of bounds in Section 4. The characteristics of the generated test problem instances are discussed in Section 5. Section 6 contains a discussion of the computational results while Section 7 presents the conclusions.

2 Literature review

The models that address the problem of convoy routing and scheduling are categorised as military mobility models.

Bovet et al. (1991) proposed a convoy scheduling problem for scheduling the movements of a collection of convoys along the same road to reach different destinations. The movements were subjected to two key constraints: each convoy has a pre-specified time-
window for its departure and convoys may not pass or cross each other along the road. Two different formulations based on integer programming and on graphs were proposed. Montana et al. (1999) investigated the problem of scheduling the move of a large amount of military equipment from a fort or a depot to a port and used genetic algorithms to solve the problem.

Baker et al. (2002) described a large scale linear programming model NRMO (NPS/Rand Mobility Optimiser) for optimising military airlift. The model routes cargo and passengers through a specified transportation network with a given fleet of aircraft subject to many physical and policy constraints. The model captures important aspects such as aerial refuelling, tactical aircraft shuttles and constraints based on crew availability. McKinzie and Barnes (2003) reviewed a number of mobility models and observed that every model uses either cumbersome ineffective classical optimisation techniques or simplistic and ineffective greedy approaches to find solutions.

Chardaire et al. (2005) gave a formal specification for the convoy movement problem and established that the corresponding feasibility problem is \(NP\)-Complete. They developed an integer programming model based on a time-space network for a simplified version of CMP and applied Lagrangean Relaxation to solve the model. They acknowledged the existence of constraints against crossing and overtaking, but simplified their model by dropping them. The authors described the use of Lagrangean Relaxation to solve the model and their results were encouraging. Tuson and Harrison (2005) demonstrated that a straightforward reformulation of the model proposed by Chardaire et al. (2005) rendered the realistic instances amenable to solution by simple heuristics. The authors stated that the \(NP\)-hardness is a worst case measure of the problem’s time complexity and real world problems may not necessarily be hard. Akgun and Tansel (2007) introduced the Deployment Planning Problem (DPP) that may be roughly defined as the problem of planning the physical movement of military units, stationed at geographically dispersed locations, from their home bases to their destinations, subject to the constraints on scheduling and routing and on various types of transportation assets. The authors proposed a mixed integer program model to solve the DPP. Gopalan and Narayanaswamy (2008) considered an online version of the CMP where convoy demands arise dynamically over time. Assuming convoys to be of zero length, i.e., particle convoys, they proposed approximation algorithms for solving the problem.

Literature is yet to provide a mathematical model that incorporates all the characteristics associated with the CMP. Some aspects of the problem have been addressed in the literature on freight transportation (Crainic and Laporte 1997, Crainic 2003), on vehicle routing and scheduling (Toth and Vigo, 2002) and on rail transportation (Cordeau et al., 1998; Higgins et al., 1996). In this paper, we develop an integer programming mathematical model for the convoy movement problem. It is an adaptation of the model proposed by Higgins et al. (1996) for scheduling trains on single track. We have also drawn ideas from Fisher (1976, 1981), Fisher and Kedia (1990), Erlenkotter (1978), Guignard (1988), Held and Karp (1970, 1971), Shapiro (1971), Beasley (1985, 1990), Schank et al. (1991) and Kress (2001) while developing the formulation.
3 Integer programming formulation

3.1 Assumptions

The following is the list of assumptions made:

- the underlying network is a bi-modal network of highways and railways consisting of nodes that represent cities and bi-directional arcs connecting them
- all convoys have equal priority
- all the nodes have sufficient infrastructure for accommodating more than one convoy at a time
- transporters are always available
- all convoys are ready at their respective home bases at time 0.

3.2 Sets and indices

\( Q \): Set of all the nodes
\( A \): Set of arcs
\( M \): Set of transportation modes
\( S \): Set of source nodes
\( D \): Set of destination nodes
\( C \): Set of military convoys need to be moved
\( K \): Very large integer.

3.3 Input parameters

\( q_s^c \): Node of convoy \( c, c \in C, q_s^c \in S \)
\( q_d^c \): Destination node of convoy \( c, c \in C, q_d^c \in D \)
\( T_{qq'} \): Number of time units required to traverse between nodes \( q \) and \( q', (q, q') \in Q \)
\( h \): Minimum headway distance to be maintained (in time units).

3.4 Variables

\( AT_q^c \): Arrival time of convoy \( c \) at node \( q \)
\( DT_q^c \): Departure time of convoy \( c \) at node \( q \)
Among the six decision variables, $AT_q^c$ and $DT_q^c$ are integer variables and the rest are binary variables. The mathematical formulation follows.

### 3.5 Objective function

The objective is to minimise the sum of arrival times of convoys at their respective destinations.

\[
\text{Minimise } \sum_c AT_q^c. \tag{1}
\]

### 3.6 Constraints

\section*{1 Network flow constraints}

This constraint ensures each convoy starts moving from its home base and reaches its destination without terminating its journey at any of the intermediary nodes.

\[
\sum_{q'} \sum_m A_{qq'}^{mc} - \sum_{q''} \sum_m A_{q''q}^{mc} = \begin{cases} 
1 & \text{if } q = q_s^c \\
0 & \text{if } q \neq q_s^c \text{ or } q_d^c \\
-1 & \text{if } q = q_d^c
\end{cases} \quad \forall c \in C, m \in M, (q, q', q'') \in Q, q \neq q' \text{ and } q \neq q''. \tag{2}
\]
2 Travel time constraints

This set of constraints ensures that each convoy takes the stipulated time to travel between any pair of nodes. In addition, it ensures that the arrival time of a convoy at its destination is never less than the sum of individual arc travel times.

\[
\begin{align*}
AT_q^c + K(1 - A_q^{mc}) & \geq T_q^m A_q^{mc} + DT_q^c, \\
AT_q^c - K(1 - A_q^{mc}) & \leq T_q^m A_q^{mc} + DT_q^c, \\
AT_q^c & \geq \sum_{q} \sum_{q'} T_q^m A_q^{mcq'}
\end{align*}
\]
\[\forall c \in C, m \in M, (q, q') \in Q \text{ and } q \neq q'. \quad (3)\]

3 No-overtaking constraints

This set of constraints ensures that when two convoys travel along the same arc in the same direction, a minimum headway time is always maintained between them.

\[
\begin{align*}
KX_{qq'}^{mcq'} + K(1 - A_{qq'}^{mcq'}) + K(1 - A_{qq'}^{mcq'}) + AT_{q}^{c} & \geq AT_{q'}^{c} + h(1 - X_{qq'}^{mcq'}) \\
K(1 - X_{qq'}^{mcq'}) + K(1 - A_{qq'}^{mcq'}) + AT_{q}^{c} & \geq AT_{q'}^{c} + hX_{qq'}^{mcq'}
\end{align*}
\]
\[\forall (q, q') \in Q, c \neq c', q \neq q' \text{ and } c' > c. \quad (4)\]

4 No-crossing constraints

This set of constraints ensures no two convoys cross each other on an arc while travelling along the same arc in opposite direction.

\[
\begin{align*}
KY_{qq'}^{mcq'} + K(1 - A_{qq'}^{mcq'}) + K(1 - A_{qq'}^{mcq'}) + DT_{q}^{c} & \geq AT_{q}^{q'} \\
K(1 - Y_{qq'}^{mcq'}) + K(1 - A_{qq'}^{mcq'}) + DT_{q}^{q'} & \geq AT_{q}^{q'}
\end{align*}
\]
\[\forall (c, c') \in C, m \in M, (q, q') \in Q, c \neq c', q \neq q' \text{ and } c' > c. \quad (5)\]

5 No-stoppage constraint

This constraint ensures that once a convoy starts from its home base, it does not stop en-route till it reaches its destination.

\[DT_q^c - AT_q^c = 0, \quad \forall c \in C, q \notin S \text{ and } q \notin D. \quad (6)\]
6 Binding constraints

(a) For convoys traversing along an arc in the same direction

This set of constraints ensures that a minimum headway time is maintained between two convoys if and only if an arc under consideration is common to both of them.

\[ X_{qq'}^{mc'} \leq A_{qq'}^{mc} \quad \forall (c, c') \in C, m \in M, (q, q') \in Q, c \neq c', q \neq q' \text{ and } c' > c \]
\[ X_{qq'}^{mc} \leq A_{qq'}^{mc} \quad \forall (c, c') \in C, m \in M, (q, q') \in Q, c \neq c', q \neq q' \text{ and } c' > c. \] (7)

(b) For convoys traversing an arc in opposite directions

In order to decide which convoy traverses a particular arc first, the following set of constraints ensures that the arc is part of both the convoys’ routes.

\[ Y_{qq'}^{mc'} \leq A_{qq'}^{mc} \quad \forall (c, c') \in C, m \in M, (q, q') \in Q, c \neq c', q \neq q' \text{ and } c' > c \]
\[ Y_{qq'}^{mc} \leq A_{qq'}^{mc} \quad \forall (c, c') \in C, m \in M, (q, q') \in Q, c \neq c', q \neq q' \text{ and } c' > c. \] (8)

(7) The following set of constraints ensures that the convoys are ready at their respective home bases at time 0 and convoys do not move after reaching their destination.

\[ AT_{q'}^{c} = 0, \quad \forall c \in C \text{ and } q' \in S \]
\[ DT_{q'}^{c} = K, \quad \forall c \in C \text{ and } q' \in D. \] (7)

4 Lower and upper bounds

The Convoy Movement Problem is a proven \(NP\)-Hard problem. Hence, optimal solutions can be obtained for problem instances up to a certain size beyond which it may not be computationally viable. For such instances, the performance of the heuristic is evaluated with reference to the lower bound.

4.1 Lower bound

A well-known method of generating a lower bound is Linear Programming (LP) relaxation in which the integrality requirement is dropped and the resulting LP is solved exactly using a standard algorithm.
4.2 Upper bound

For the objective of minimisation, upper bounds are generated using heuristic techniques. In the present problem, conflicts arise between convoys only at two specific instances: The conflicts are resolved on a first-come-first-serve basis.

**Situation 1:** Two convoys attempt to traverse a bi-modal arc in opposite directions at the same time along same mode of transportation.

**Resolving Mechanism:** In Figure 1, let $EE'$ be a bi-modal (modes $m_1$ and $m_2$) conflicting edge. Suppose convoy $C_1$ reaches node $E$ before $C_2$ reaches node $E'$: Convoy ‘$C_1’$ is allowed to traverse arc $EE'$ using the faster mode of transport and direct convoy ‘$C_2’$ to traverse $E'E$ by the slower mode of transport.

**Situation 2:** Two convoys competing to traverse along a uni-modal arc in opposite directions at the same time using same mode of transportation.

**Resolving Mechanism:** In Figure 2, let $EE'$ be a uni-modal conflicting edge. Suppose convoy $C_2$ reaches node $E'$ before convoy $C_1$ reaches node $E$: Convoy $C_2$ is allowed to traverse arc $E'E$ and convoy $C_1$ is forbidden from traversing along arc $EE'$; accordingly, its traverse time on that arc is set to $\infty$. An alternative route is found between the same nodes using Dijkstra’s algorithm.

4.2.1 Step-by-step procedure

1. Solve the problem using Dijkstra’s shortest path algorithm.
2. Examine the obtained solution for conflicts. If there is no conflict, terminate the heuristic by reporting this solution as final. Else, proceed to step 3.
3. Resolve all the conflicts that arise using the proposed resolving mechanisms and then record the final solution.

5 Test problem instances

Since information pertaining to defence planning and logistics is classified, hypothetical problem instances were generated to evaluate the efficacy of the proposed formulation. The problems could be classified as instances with and without known optimal solutions. A total of 12 networks with problem sizes ranging from ten cities and three convoys to 100 cities and 30 convoys were generated. An Arc Density Factor of 0.15 and Node-Convoy ratio of $[3.5 - 5]$ was assumed for all the networks. These are computed as shown below:

\[
\text{Arc Density Factor (}\rho\text{)} = \frac{\text{Degree of a node}}{\text{Total number of nodes}}
\]
To evaluate the robustness of the formulation, three different scenarios were introduced for each network, using a factor called the ‘Identical Destination Factor’, defined as follows:

\[
\text{Identical Destination Factor (} \theta \text{)} = \frac{\text{Number of identical destination nodes}}{\text{Total number of convoys}}.
\]

An identical destination factor ‘0’ indicates that no more than one convoy heads towards any destination; \( \theta = 1 \) indicates that all the convoys head towards a common destination.

### 6 Results and discussions

The models were coded in C++ language and solved using ILOG CPLEX 9.0 Concert Technology. The computational times reported are in CPU seconds on a Zenith PC working on Intel Pentium 4, 3 GHz processor with 1 GB of RAM. All the results obtained are summarised in Tables 1–3. Tables 1 and 2 pertain to problem instances for which CPLEX could give optimal solutions. Table 3 pertains to problem instances without optimal solutions.

Table 1 reports the Best case performance, Worst case performance and Average case performance of the formulation in terms of average CPU time and average number of iterations up to a problem size of 50 cities and ten convoys. There does not appear to be a significant difference between best case and worst case performances for smaller problems. As the problem size increases, the difference in performance grows exponentially. Figure 3 depicts a graph plotted between Network size and Average CPU time for \( \theta = 0.5 \).

In Table 2, the quality of the LP bounds and upper bounds are reported with respect to the optimal solutions obtained. In Table 3, for problem instances without optimal solutions, we have reported the performance of the proposed heuristic evaluated with respect to the lower bound for four large networks with 15 instances each.

### 7 Conclusions

The strategic problem of routing military convoys between specific origin and destination pairs within a network has been considered in this study. Formulated as an integer linear program, it is found amenable to optimisation for problem-sizes up to 50 cities and ten convoys, in terms of the amount of computational time required. Since it is not possible to find optimal solutions to larger problems, the proposed heuristic is evaluated against the lower bound for such instances. Viewed against the encouraging results obtained in this paper, the proposed model marks a significant improvement over the existing models in the literature.
References


List of Tables

Table 1  Average CPU times and average number of iterations
Table 2  Results for problem instances with optimal solutions
Table 3  Results for problem instances without optimal solutions

List of Figures

Figure 1  Convoys competing to traverse a bi-modal arc
Figure 2  Convoys competing to traverse a uni-modal arc
Figure 3  Network size vs. average CPU time for $\theta = 0.5$ (see online version for colours)
<table>
<thead>
<tr>
<th>Network</th>
<th>No. of nodes</th>
<th>No. of convoys</th>
<th>Identical destination factor (θ)</th>
<th>No. of problems solved</th>
<th>Best case performance</th>
<th>Worst case performance</th>
<th>Average performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CPU time</td>
<td>No. of iterations</td>
<td>CPU time</td>
<td>No. of iterations</td>
</tr>
<tr>
<td>N1</td>
<td>10</td>
<td>3</td>
<td>0.0</td>
<td>10</td>
<td>0.09</td>
<td>8</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>0.17</td>
<td>9</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>0.38</td>
<td>35</td>
<td>1.66</td>
</tr>
<tr>
<td>N2</td>
<td>15</td>
<td>4</td>
<td>0.5</td>
<td>10</td>
<td>0.37</td>
<td>34</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>0.58</td>
<td>45</td>
<td>2.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>0.65</td>
<td>85</td>
<td>4.68</td>
</tr>
<tr>
<td>N3</td>
<td>20</td>
<td>5</td>
<td>0.5</td>
<td>10</td>
<td>0.82</td>
<td>73</td>
<td>8.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>0.88</td>
<td>86</td>
<td>7.99</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>1.74</td>
<td>114</td>
<td>33.92</td>
</tr>
<tr>
<td>N4</td>
<td>25</td>
<td>6</td>
<td>0.5</td>
<td>10</td>
<td>2.36</td>
<td>130</td>
<td>26.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>1.64</td>
<td>106</td>
<td>42.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>4.85</td>
<td>193</td>
<td>70.80</td>
</tr>
<tr>
<td>N5</td>
<td>30</td>
<td>7</td>
<td>0.5</td>
<td>10</td>
<td>13.32</td>
<td>314</td>
<td>60.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>17.88</td>
<td>339</td>
<td>121.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>6.60</td>
<td>229</td>
<td>284.5</td>
</tr>
<tr>
<td>N6</td>
<td>35</td>
<td>8</td>
<td>0.5</td>
<td>10</td>
<td>6.47</td>
<td>226</td>
<td>308.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>8.66</td>
<td>206</td>
<td>274.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>29.60</td>
<td>447</td>
<td>255.52</td>
</tr>
<tr>
<td>N7</td>
<td>40</td>
<td>9</td>
<td>0.5</td>
<td>10</td>
<td>37.99</td>
<td>465</td>
<td>536.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>42.46</td>
<td>413</td>
<td>824.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.0</td>
<td>10</td>
<td>29.33</td>
<td>575</td>
<td>297.93</td>
</tr>
<tr>
<td>N8</td>
<td>50</td>
<td>10</td>
<td>0.5</td>
<td>10</td>
<td>42.21</td>
<td>624</td>
<td>723.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.0</td>
<td>10</td>
<td>44.34</td>
<td>592</td>
<td>1204.61</td>
</tr>
</tbody>
</table>

Table 1
<table>
<thead>
<tr>
<th>Problem size</th>
<th>No. of problem instances solved</th>
<th>Average quality of LP bound</th>
<th>Average quality of Heuristic solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 × 3</td>
<td>30</td>
<td>0.51</td>
<td>1.39</td>
</tr>
<tr>
<td>15 × 4</td>
<td>30</td>
<td>0.92</td>
<td>2.04</td>
</tr>
<tr>
<td>20 × 5</td>
<td>30</td>
<td>1.70</td>
<td>2.83</td>
</tr>
<tr>
<td>25 × 6</td>
<td>30</td>
<td>2.23</td>
<td>3.94</td>
</tr>
<tr>
<td>30 × 7</td>
<td>30</td>
<td>3.41</td>
<td>5.88</td>
</tr>
<tr>
<td>35 × 8</td>
<td>30</td>
<td>3.72</td>
<td>5.29</td>
</tr>
<tr>
<td>40 × 9</td>
<td>30</td>
<td>5.29</td>
<td>6.50</td>
</tr>
<tr>
<td>50 × 10</td>
<td>30</td>
<td>5.84</td>
<td>8.05</td>
</tr>
</tbody>
</table>

*Quality of LP bound = ((Optimal solution – LP bound)/Optimal solution) × 100%.

*Quality of Heu. Solution = (Heu.soln – Optimal solution)/Optimal solution × 100%.
<table>
<thead>
<tr>
<th>Problem size</th>
<th>No. of problem instances solved</th>
<th>Performance of the proposed heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>$60 \times 15$</td>
<td>15</td>
<td>13.38</td>
</tr>
<tr>
<td>$75 \times 20$</td>
<td>15</td>
<td>12.66</td>
</tr>
<tr>
<td>$85 \times 25$</td>
<td>15</td>
<td>13.51</td>
</tr>
<tr>
<td>$100 \times 30$</td>
<td>15</td>
<td>15.68</td>
</tr>
</tbody>
</table>

*% gap = (Heuristic solution – LP bound)/LP bound) × 100%.*

Table 3
Figure 2
Figure 3